

General Task:

I will build a system that pulls tweets from the Twitter API based on a search term. I'm initially planning on doing something election-related but I'm going to wait until I have the system set up to select a term so I can be sure my system is not overwhelmed. I will pull the Tweets using the Camel interface with the Twitter API and convert them to a MESSAGE that will pass through a CONTENT BASED ROUTER on a BROKER that will segment them based on different factors. This will then become multiple PUBLISH-SUBSCRIBE Channels that MESSAGE ENDPOINTS will subscribe too. The content-based routers will be determined by different terms likely candidate names (Trump, Clinton and Sanders) and the different subscribers will pull the Twitter messages from those channels.

Each subscriber will be built off the same TEMPLATE METHOD and there will be a separate method for each candidate. These templates will pull out relevant information from the messages, some using candidate specific rules and others using more general STRATEGIES that are dictated for the system as a whole with rules on how to aggregate data as it comes off the message queues.

Analysis:

Once the data is taken off of the queue it will be analyzed in multiple LAYERS that will interact with each other through FACADES. These layers will provide multiple analysis functions and at the end will eventually save a file that provides summary information about the set of tweets that came in. The first layer will standardize the data and convert it into hashes that can be converted into columns. The second will consolidate the data into a data structure, potentially a database or just a set of combined hashes. It will then activate an OBSERVER from the final layer that will see if the data is actually new and will save the new data to a file tagged with the strategy used.

Deliverable:

The final system will likely have the following classes, each in their own class:

- TwitterAPI
- MessageRouter
- MessageRecievers
- StrategyBase
- Strategies (*3)
- Summerizer
- SummerizerFacade
- Analyzer
- AnalyzerHelper
- AnalyzerFacade
- AnalyzeObserver

- FileWriter

Big Picture:

